

Chapter 13: Network Programs Available on Kerberized Machines

In this chapter we document the Kerberized features of the network connection programs that are usable with Kerberos v5.

13.1 Introduction

The Kerberos V5 network programs are versions of existing UNIX network programs with the Kerberos features added. We call these versions “Kerberized”. They include **telnet**, **rsh**, **rlogin**, **FTP**, and **rcp** which come with the installation of a Kerberos 5 client, and **ssh**, **slogin** and **scp** which come with a Kerberized **ssh** client. These programs have the original features of the corresponding non-Kerberized programs, plus additional features that transparently use your Kerberos tickets for negotiating authentication and optional encryption with the remote host. In most cases, all you’ll notice is that you no longer have to type your password, because Kerberos has already proven your identity.



Be aware that, depending on how the network program is configured and whether the target machine is Kerberized, you may be prompted for either your login id or password, both, or neither.

You can check the defaults set for the (non-ssh) programs in the [appdefaults] section of the `/etc/krb5.conf` file. For **ssh** configuration, see the **ssh** man pages. These defaults can be overridden via command line options (and in the cases of **telnet** and **FTP** when invoked without a hostname argument, via commands inside the program).

In this chapter we list only the command syntax and the Kerberos-added features for these programs.

13.2 Kerberized telnet

Communicate with another host using the TELNET protocol. Use with a host argument to open a connection to that host.

```
% telnet [-8] [-E] [-F] [-K] [-L] [-N] [-S <tos>] \
  [-X <authtype>] [-a] [-c] [-d] [-e <escapechar>] [-f] \
  [-k <REALM>] [-l <user>] [-n <tracefile>] [-r] [-x] \
  [<host> [<port>]]
```

The following are the Kerberos options:

- a** attempts automatic login using your tickets. **telnet** will assume you want the same login id on the remote host unless you explicitly specify another (using **-l**).
- f** forwards a copy of your existing tickets to the remote host, but does not mark them as reforwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files.
- F** forwards a copy of your existing tickets to the remote host, and marks them as re-forwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files.
- k <REALM>** requests tickets in the specified realm, which may be different from the one the system would use by default.
- K** uses your tickets to authenticate to the remote host, but does not log you in; i.e., specifies "no auto-login".
- N** turns off ticket forwarding to the remote system.

Use of this option overrides any forwarding defaults specified in your machine's configuration files.
- x** (encrypt) turns on encryption.

Use of this option overrides any encryption defaults specified in your machine's configuration files.
- X <atype>** disable **atype** type of authentication

Example:

Log in to the remote Kerberized machine `fsgi03.fnal.gov`, assume your username is different on this machine (**-l qsmith**). Forward tickets and mark them as reforwardable from the target machine (**-F**):

```
% telnet -F -l qsmith fsgi03.fnal.gov
```

13.3 Kerberized rsh

Connect to a specified host, and execute a specified command on that host.

```
% rsh <host> [-l <login_name>] [-n] [-d] [-k <REALM>] [-f | -F]
  \ [-N] [-x] [-X] <command>
```

If **<command>** is left off, **rsh** runs **rlogin**.

The following are the Kerberos options:

- | | |
|-------------------------|--|
| -f | forwards a copy of your existing tickets to the remote host, but does not mark them as reforwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files. |
| -F | forwards a copy of your existing tickets to the remote host, and marks them as re-forwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files. |
| -k <REALM> | requests tickets in the specified realm, which may be different from the one the system would use by default. |
| -n | This is not a Kerberos option, but we include it with a usage note. As in non-Kerberized rsh, -n redirects input from the special device <code>/dev/null</code> . If you put a command rsh <host> <command> in the background with & , it will stop because only a foreground process can access the tty for input. If you make it rsh -n <host> <command> , the rsh command does not have the tty open for input at all, so it does not get stopped. |
| -N | turns off ticket forwarding to the remote system.

Use of this option overrides any forwarding defaults specified in your machine's configuration files. |
| -x | (encrypt) turns ON encryption for the session

Use of this option overrides any encryption defaults specified in your machine's configuration files. |
| -X | turns OFF encryption of the session.

Use of this option overrides any encryption defaults specified in your machine's configuration files. |

Example:

Run the command **date** on the remote Kerberized machine `fsui03.fnal.gov`, and assume your username is different on it (**-l qsmith**). The command doesn't require Kerberos tickets in order to run, nor does it require encryption (**-X** turns it off):

```
% rsh fsgi03.fnal.gov -l qsmith -X date
```

13.4 Kerberized rlogin

Log into a remote host. Kerberos authentication is used in place of the rhosts mechanism to determine if user is authorized to use remote account.

```
% rlogin <rhost> [-e<c>] [-8] [-c] [-a] [-f] [-F] [-N] \
  [-t <termtype>] [-n] [-7] [-noflow] [-d] [-k <REALM>] [-x]\
  [-X] [-L] [-l <username>]
```

The following are the Kerberos options:

- | | |
|-------------------------|---|
| -f | forwards a copy of your existing tickets to the remote host, but does not mark them as reforwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files. |
| -F | forwards a copy of your existing tickets to the remote host, and marks them as re-forwardable from there.

Use of this option overrides any forwarding defaults specified in your machine's configuration files. |
| -k <REALM> | requests tickets in the specified realm, which may be different from the one the system would use by default. |
| -N | turns off ticket forwarding to the remote system.

Use of this option overrides any forwarding defaults specified in your machine's configuration files. |
| -x | (encrypt) turns ON encryption for the session

Use of this option overrides any encryption defaults specified in your machine's configuration files. |
| -X | turns OFF encryption of the session.

Use of this option overrides any encryption defaults specified in your machine's configuration files. |

Example:

Log into the remote Kerberized machine `fsui03.fnal.gov`, assume your username is different on it (`-l qsmith`), forward a reforwardable copy of the local Kerberos credentials (`-F`):


```
% rlogin fsgi03.fnal.gov -l qsmith -F
```

13.5 Kerberized FTP

Transfer files to and from a remote host. FTP prompts the user for a command. Type **help** to see a list of commands.

```
% ftp [-v] [-d] [-i] [-n] [-g] [-k <REALM>] [-f] [-x] [-u] [-t]\
    [<host>]
```

The following are the Kerberos options:

- 
- f** requests that your tickets be forwarded to the remote host. (This is necessary if the remote host runs AFS.)
 - k <REALM>** Ignore this option of the ftp client. It has nothing to do with Kerberos v5. It does work for telnet and the r-commands.
 - n** no auto-login attempt at initial connection, but still does Kerberos authentication
- protect <level>**(issued at the **ftp>** prompt) sets the protection level. The level **clear** is “no protection”; **safe** ensures data integrity, and **private** encrypts the data and ensures data integrity.
- u** restrains **FTP** from attempting auto-authentication; also disables auto-login.

Note: If your local and remote login names don’t match, you can enter your login name for the remote system at the prompt that you get after you issue the **ftp** command.

Examples:

Transfer files from a remote nonKerberized machine `www.xyz.org`, and assume your username is different on it:

```
% ftp www.xyz.org
Connected to xyz.org.
220 ...
500 AUTH not understood.
KERBEROS_V4 rejected as an authentication type
```

```

Name (www.xyz.org:aheavey): anneh
331 Password required for anneh.
Password:
230 User anneh logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
-rw-rw-r-- 1 batavia site23      1700 Jan 25 10:52 header_1.GIF
...
ftp> get header_1.GIF
local: header_1.GIF remote: header_1.GIF
200 PORT command successful.
150 Opening BINARY mode data connection for header_1.GIF (1700 bytes).
226 Transfer complete.
1700 bytes received in 0.016 seconds (1e+02 Kbytes/s)
ftp> bye
221 Goodbye.

```

Transfer files from a remote Kerberized machine `abc.minos-soudan.org` that runs AFS (you must forward credentials, **-f**). Assume your username is different on each machine. Set the protection to “private” in order to encrypt the data and ensure data integrity:

```
% ftp -f abc.minos-soudan.org
```

```

Connected to abc.minos-soudan.org.
...
220 abc.minos-soudan.org FTP server (Version 5.60) ready.
334 Using authentication type GSSAPI; ADAT must follow
GSSAPI accepted as authentication type
GSSAPI authentication succeeded
Name (abc.minos-soudan.org:aheavey): crluser
232 GSSAPI user aheavey@FNAL.GOV is authorized as crluser
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> private
200 Data channel protection level set to private.
ftp> get lmnop.qrs
...
ftp> bye
221 Goodbye.

```

13.6 Kerberized rcp

Copy files between machines. Each file or directory argument is either a remote file name of the form `remote_host:path` or a local file name/path.

```
% rcp [-p] [-x] [-X] [-k <REALM>] [-D <port>] [-n] [-F] [-N] \
  [-c <cache>] [-C <config>] <file1> <file2>
```

or

```
% rcp [-p] [-x] [-X] [-k <REALM>] [-r] [-D <port>] [-n] [-F] \
  [-N] [-c <cache>] [-C <config>] <file> ... <directory>
```



The following are the Kerberos options:

- c <cache>** uses credentials file **<cache>** instead of default
- F** forwards credentials to remote system (This is needed if the other end runs AFS.)
- k <REALM>** requests tickets for the remote host in the specified realm, which may be different from the one the system would use by default.
- N** turns off ticket forwarding to the remote system.
Use of this option overrides any forwarding specified in your machine's configuration files.
- x** (encrypt) turns on encryption.
- X** turns off encryption of the session.
Use of this option overrides any encryption specified in your machine's configuration files.

Examples:

Copy the local files `def.histo` and `ghi.histo` to your home directory on the remote machine `jkl.myuniv.edu`. Assume the remote machine does not run AFS. Your username is the same on both:

```
% rcp def.histo ghi.histo jkl.myuniv.edu:
```

Copy the local directory `histo` and all subdirectories to your home directory on the remote machine `jkl.myuniv.edu`. Assume the remote machine does not run AFS. Your username is the same on both:

```
% rcp /path/to/histo jkl.myuniv.edu:
```

Copy all the files from the directory `/path/to/mno` on the remote node `pqr.myuniv.edu` into the local directory `~stu/vwx` (quote the first argument to prevent filename expansion from occurring on the local machine):

```
% rcp "pqr.myuniv.edu:/path/to/mno/*" ~stu/vwx
```

13.7 Kerberized su (ksu)

Be aware that you need to have a host principal in order to use `ksu`. See section 14.1.6 *Do you Need to Allow Incoming Kerberos Connections?* about host principals. The following discussion is adapted from the **ksu** man pages. See them for more information, in particular for option descriptions. The command syntax is:

```
% ksu [<target_user>] [-n <target_principal_name>] [-c \
<source_cache_name>] [-C <target_cache_name>] [-k] [-D] [-r \
<time>] [-pf] [-l <lifetime>] [-zZ] [-q] [-e <command> \
[<args ...>]] [-a [<args ...>]]
```

The Kerberos V5 **ksu** program is a Kerberized version of the **su** program that has two missions: one is to securely change the real and effective user ID to that of the target user, the other is to create a new security context.

To fulfill the first mission, **ksu** operates in two phases: authentication and authorization. Resolving the target principal name is the first step in authentication. If the source user is *root* or the target user is the source user, no authentication or authorization takes place. In all other cases, **ksu** looks for an appropriate Kerberos ticket in the source cache. If no ticket is in the cache, then depending on how **ksu** was compiled, the user may be prompted for a Kerberos password.



Make sure you are logged in using an encrypted connection before typing your password!

Upon successful authentication, **ksu** checks whether the target principal is authorized to access the target account. In the target user's home directory, authorization is based on whether appropriate entries exist in either *.k5login* or *.k5users*, or by name-mapping rules if neither file exists.

ksu can be used to create a new security context for the target program. The target program inherits a set of credentials from the source user. By default, this set includes all of the credentials in the source cache plus any additional credentials obtained during authentication. The source user is able to limit the credentials in this set.

13.8 Kerberized ssh and slogin

The **ssh** and **slogin** commands are intended to replace **rsh** and **rlogin** (see sections 13.3 *Kerberized rsh* and 13.4 *Kerberized rlogin*) and to provide secure encrypted connections between two untrusted hosts over an insecure network. If the **<command>** argument is left off, **ssh** runs **slogin**.

```
% ssh [-a] [-c idea|blowfish|des|3des|arcfour|none] \
[-e <escape_char>] [-i <identity_file>] [-l <login_name>]\
[-n] [-k] [-V] [-o <option>] [-p <port>] [-q] [-P] [-t] [-v]\
[-x] [-C] [-g] [-L <port>:<host>:<hostport>] [-R \
<port>:<host>:<hostport>] <hostname> [<command>]
```

-c Specifies cipher for encrypting connection; not needed if specified in configuration file

-k Disables forwarding of the kerberos tickets. This may also be specified on a per-host basis in the configuration file.

Any Kerberos options would be used within **-o <ssh-options>**.

Examples:

From your local machine, log into the remote node fsgi03.fnal.gov on which your (different) username is qsmith. Respond **yes** if asked if you want to continue:

```
% slogin fsgi03.fnal.gov -l qsmith
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)?
```

From your local machine, run the **date** command on the remote node fsgi03.fnal.gov, but don't start a session:

```
% ssh fsgi03.fnal.gov -l qsmith date
```

13.9 Kerberized scp

Copy files between hosts on a network, using ssh for data transfer.

```
% scp [-a] [-A] [-q] [-Q] [-p] [-r] [-v] [-B] [-C] [-L] [-l] \  
  [-S <path_to_ssh>] [-o <ssh-options>][-P <port>] \  
  [-c idea|blowfish|des|3des|arcfour|none] [-i <identity>]\  
  [[user@host1:]filename1... [user@host2:]filename2]
```

Any Kerberos options would be used within **-o <ssh-options>**.

Example:

Log into a Kerberized machine at Fermilab, and pull files from a remote machine, mynode.myuniv.edu. On the remote node the username is qsmith, and on the local node, it's quentins. The user wants to pull a file from mynode.myuniv.edu to his local Fermilab machine:

```
% scp qsmith@mynode.myuniv.edu:/home/qsmith/muonrun47.histo \  
  ~quentins/geant4/work/muonhistos
```

